



Міністерство освіти і науки України  
Національний технічний університет України “Київський політехнічний  
інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

Лабораторна робота №2  
З дисципліни «Технології розроблення програмного забезпечення»  
Тема: **«ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЙ  
ВАРІАНТІВ ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ.  
КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ»**  
Powershell terminal

Виконав:  
Студент групи ІА-22  
Парій І. Р.

Перевірив:  
Мягкий М. Ю.

Київ-2024

## **Зміст**

Тема:.....	3
Мета:.....	3
Хід роботи.....	3
1. Схема прецедентів.....	3
2. Деталі сценаріїв використання.....	4
3. Схема класів.....	5
4. Структура бази даних.....	7
Висновки:.....	8

## Тема:

Діаграма варіантів використання. Сценарії варіантів використання. Діаграми uml. Діаграми класів. Концептуальна модель системи

## Мета:

Вивчити та застосувати UML-діаграми для моделювання варіантів використання та концептуальної структури даних системи, з акцентом на діаграмах класів, прецедентів та розробці сценаріїв їхнього використання.

## Хід роботи

### 1. Схема прецедентів

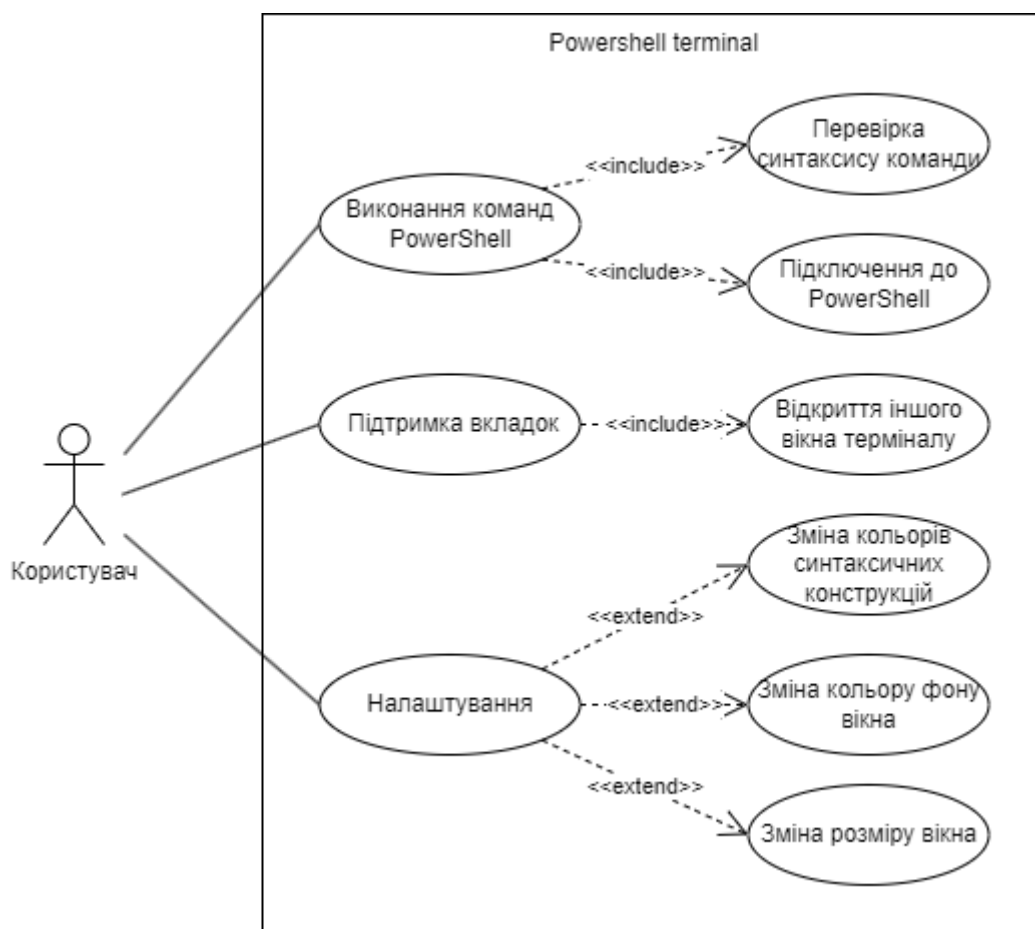


Рис. 1 - Схема прецедентів

## 2. Деталі сценаріїв використання

### Сценарій використання 1: Виконання команди PowerShell

**Передумови:** Користувач має доступ до терміналу PowerShell та знає потрібну команду.

**Післяумови:** Команда виконується, і результат виводиться в терміналі.

**Актори:** Користувач.

**Опис:** Цей сценарій описує процес виконання команди PowerShell у терміналі.

**Основний хід подій:**

1. Користувач відкриває PowerShell термінал.
2. Користувач вводить команду PowerShell.
3. Система перевіряє синтаксис введеної команди.
4. Якщо синтаксис правильний, команда виконується.
5. У терміналі відображається результат виконання команди.

**Винятки:** Якщо синтаксис команди неправильний, система видає повідомлення про помилку.

### Сценарій використання 2: Підтримка вкладок

**Передумови:** Користувач працює в PowerShell терміналі та бажає відкрити додаткове вікно для одночасної роботи з іншими командами.

**Післяумови:** Відкрите нове вікно або вкладка терміналу.

**Актори:** Користувач.

**Опис:** Цей сценарій описує процес відкриття іншого вікна або вкладки терміналу.

**Основний хід подій:**

1. Користувач відкриває меню терміналу або використовує гарячі клавіші для створення нової вкладки.
2. Система створює нову вкладку або нове вікно терміналу.
3. У новому вікні терміналу користувач може вводити команди незалежно від основного вікна.

**Винятки:** Якщо система не може відкрити нову вкладку (наприклад, через перевантаження), користувач отримує повідомлення про помилку.

### Сценарій використання 3: Налаштування кольорів синтаксичних конструкцій

**Передумови:** Користувач має доступ до меню налаштувань PowerShell терміналу.

**Післяумови:** Кольори синтаксичних конструкцій змінені, нові налаштування застосовуються.

**Актори:** Користувач.

**Опис:** Цей сценарій описує процес зміни кольорів синтаксичних конструкцій у PowerShell терміналі.

**Основний хід подій:**

1. Користувач відкриває меню налаштувань терміналу.
2. У меню налаштувань користувач переходить до розділу кольорів синтаксичних конструкцій.
3. Користувач вибирає нову колірну схему або налаштовує кольори вручну.
4. Система зберігає зміни та застосовує нові налаштування до всіх майбутніх команд.

**Виятки:** Відсутні.

#### 3. Схема класів

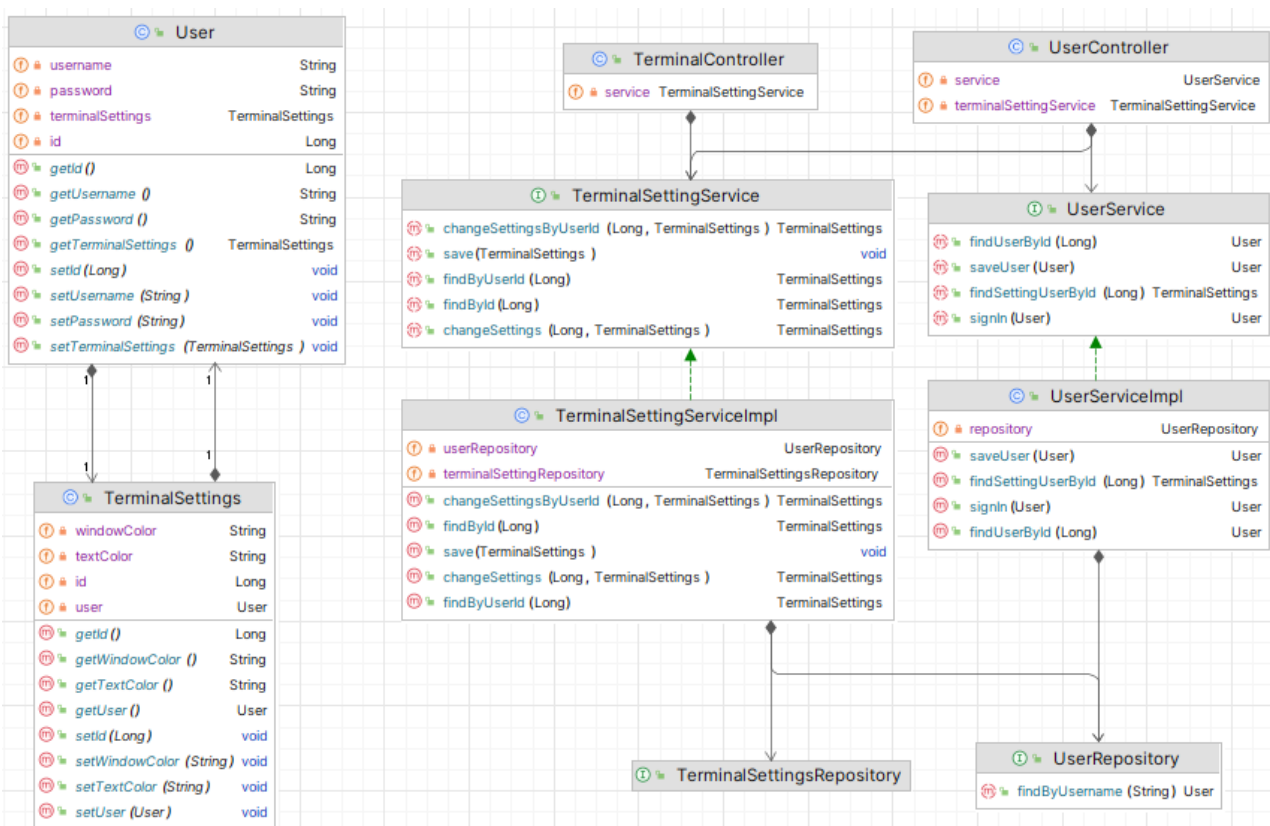


Рис. 2 - Схема класів

## **1. User**

- Призначення: Представляє користувача системи. Зберігає дані про користувача, такі як ім'я, пароль і його індивідуальні налаштування терміналу.
- Роль у системі: Цей клас використовується для аутентифікації користувачів, прив'язки їхніх налаштувань терміналу та управління їх обліковими записами.

## **2. TerminalSettings**

- Призначення: Зберігає налаштування терміналу для кожного користувача.
- Роль у системі: Цей клас відповідає за зберігання параметрів, таких як кольори вікна і тексту, що дозволяє кастомізувати вигляд терміналу відповідно до потреб користувача.

## **3. UserController**

- Призначення: Обробляє запити, пов'язані з управлінням користувачами (наприклад, реєстрація, вхід у систему).
- Роль у системі: Цей контролер є точкою входу для запитів, що стосуються користувачів, і викликає сервіси для виконання відповідних операцій.

## **4. TerminalController**

- Призначення: Обробляє запити, пов'язані з управлінням налаштуваннями терміналу.
- Роль у системі: Забезпечує доступ до функцій зміни налаштувань терміналу для користувача, наприклад, зміна кольорів тексту чи вікна.

## **5. UserService**

- Призначення: Описує бізнес-логіку, пов'язану з користувачами (збереження, пошук, вхід).
- Роль у системі: Сервісний шар для виконання операцій над користувачами. Реалізується через UserServiceImpl.

## **6. UserServiceImpl**

- Призначення: Реалізує методи, визначені в UserService, і взаємодіє з базою даних через UserRepository.
- Роль у системі: Реалізує основну логіку роботи з користувачами, включаючи їх збереження, пошук та авторизацію.

## **7. TerminalSettingService**

- Призначення: Описує бізнес-логіку для роботи з налаштуваннями терміналу (зміна, збереження, пошук).

- Роль у системі: Сервісний шар для виконання операцій із налаштуваннями терміналу. Реалізується через TerminalSettingServiceImpl.

#### 8. TerminalSettingServiceImpl

- Призначення: Реалізує методи, визначені в TerminalSettingService, і взаємодіє з базою даних через TerminalSettingsRepository.
- Роль у системі: Виконує зміну, збереження та пошук налаштувань терміналу, зокрема для конкретних користувачів.

#### 9. UserRepository

- Призначення: Забезпечує доступ до даних користувачів у базі.
- Роль у системі: Дозволяє зберігати, оновлювати та отримувати інформацію про користувачів.

#### 10. TerminalSettingsRepository

- Призначення: Забезпечує доступ до даних налаштувань терміналу в базі.
- Роль у системі: Дозволяє зберігати, оновлювати та отримувати інформацію про налаштування терміналу.

### 4. Структура бази даних

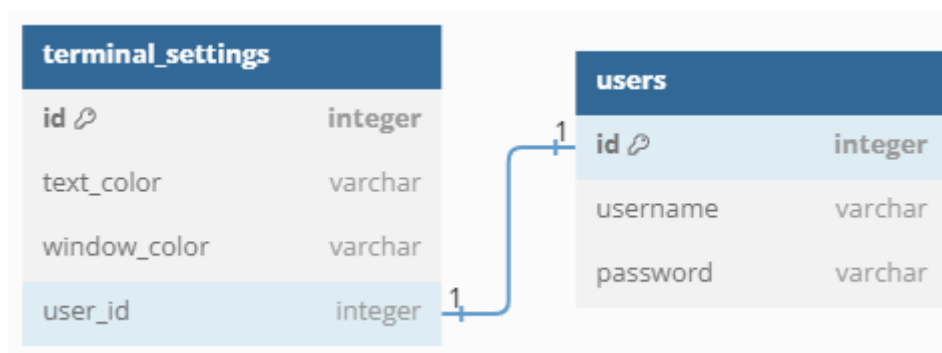


Рис. 3 - Структура бази даних

Зображена структура бази даних складається з двох таблиць: terminal\_settings і users. Вони взаємопов'язані зв'язком "один до одного" через ключове поле user\_id.

#### Таблиця users:

Ця таблиця зберігає дані користувачів системи. Поля:

- id (integer): первинний ключ таблиці (унікальний ідентифікатор користувача).

- username (varchar): ім'я користувача (логін).
- password (varchar): пароль користувача.

### **Таблиця terminal\_settings:**

Ця таблиця містить налаштування терміналу для кожного користувача. Поля:

- id (integer): первинний ключ таблиці.
- text\_color (varchar): колір тексту, який відображається в терміналі.
- window\_color (varchar): колір фону вікна терміналу.
- user\_id (integer): зовнішній ключ, що пов'язує цю таблицю з таблицею users. Забезпечує відношення "один до одного" між налаштуваннями терміналу та користувачем.

### **Зв'язок між таблицями:**

- Кожен користувач з таблиці users може мати один запис у таблиці terminal\_settings.
- Поле user\_id у terminal\_settings посилається на поле id у users.

### **Призначення:**

- Ця структура дозволяє:
- Зберігати індивідуальні налаштування терміналу (кольори тексту, фону тощо) для кожного користувача.
- Забезпечувати аутентифікацію користувачів через таблицю users.
- Використовувати налаштування для відображення терміналу у відповідності до переваг користувача.

### **Висновки:**

Вивчено та застосовано UML-діаграми для моделювання варіантів використання та концептуальної структури даних системи, з особливою увагою до діаграм класів, прецедентів і розробки відповідних сценаріїв їхнього використання.